CLAIMS:

What is claimed is:

1. A method, comprising:

registering, in response to an instruction included in source code for an upstream component, a procedure at a downstream component in a packet processing pipeline, the procedure being associated with at least one event;

processing a received packet at the upstream component executing on a first engine;

processing the packet at the downstream component executing on a second engine after the processing of the received packet at the upstream component, the processing at the downstream component comprising:

determining occurrence of the at least one associated event at the downstream component; and

in response, executing the registered procedure at the second engine.

2. The method of claim 1, wherein the first engine and second engine comprise engines integrated on the same semiconductor die.

3. The method of claim 2, wherein the first engine and the second engine comprise multi-threaded engines.

4. The method of claim 1, wherein the procedure comprises a procedure that alters data structures defined by the upstream component.

5. The method of claim 1, wherein the upstream component and downstream component comprise one of: adjacent components in the pipeline and non-adjacent components in the pipeline.

6. The method of claim 1, wherein the registering comprises one of: run-time registering and compile-time registering.

7. The method of claim 1, wherein the registering comprises loading instructions for the procedure into the second engine.

8. The method of claim 7, wherein the registering comprises registering the procedure with an event handler that invokes registered procedures in response to events signaled by the downstream component.

9. The method of claim 1, wherein the packet processing pipeline comprises one of the following: an IPv4 packet processing pipeline, an IPv6 packet processing pipeline, and an Asynchronous Transfer Mode (ATM) packet processing pipeline.

10. A computer program, disposed on a computer readable medium, the program comprising instructions to cause a processor to:

access instructions of an upstream component in a packet processing pipeline, the upstream component to be executed by a first engine;

register, in response to an accessed instruction of the upstream component, a procedure at a downstream component in the packet processing pipeline to be executed by a second engine, the procedure to be executed on the second engine in response to the downstream component detecting the occurrence of at least one event.

11. The program of claim 10, wherein the first engine and second engine comprise engines integrated on the same semiconductor die.

12. The program of claim 11, wherein the first engine and the second engine comprise multi-threaded engines.

13. The program of claim 10,

wherein the procedure comprises a procedure that alters data structures defined by the upstream component.

14. The program of claim 10, wherein the upstream component and downstream component comprise one of: adjacent components in the pipeline and non-adjacent components in the pipeline.

15. The program of claim 10, wherein the program comprises a compiler.

16. The program of claim 10, wherein the instructions to register comprise instructions to load instructions for the procedure into the second engine.

17. The program of claim 16, wherein the instructions to register comprises instructions to register the procedure with an event handler that invokes registered procedures in response to events signaled by the downstream component.

18. The program of claim 10, wherein the packet processing pipeline comprises one of the following: an IPv4 packet processing pipeline, an IPv6 packet processing pipeline, and an Asynchronous Transfer Mode (ATM) packet processing pipeline.